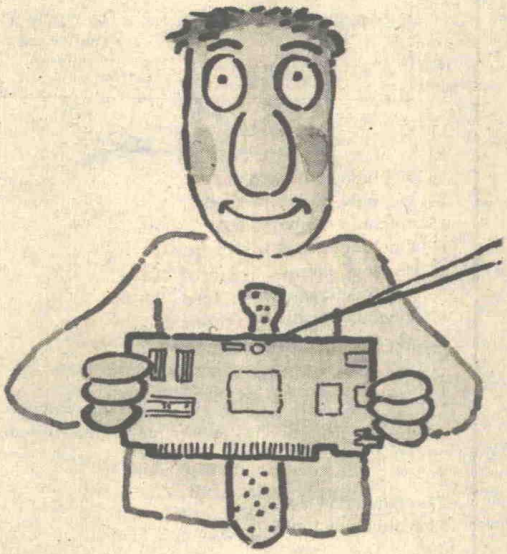
ภาษา C

จเร เลิศสุดวิชัย

SOFT MAIL

พัฒนาโปรแกรมยูทิลิตี้ด้วย เทอร์โบซี ตอนที่ 5

นำเสนอโปรแกรมยูทิลิตี้ที่ใช้รักษาความปลอดภัยข้อมูล และการเข้ารหัสข้อมูล รวม 3 โปรแกรม



โปรแกรมยูทิลิตี้ทั้งสามโปรแกรมในมุมภาษาซีตอนนี้เป็นโปรแกรมที่เกี่ยวกับการรักษาความปลอดภัยข้อมูล โดยโปรแกรมทั้งสามนี้มีการทำงานที่เรียบง่าย ซึ่งเป็นลักษณะเดียวกับโปรแกรมยูทิลิตี้ที่เคยเสนอไปแล้ว โปรแกรมแรกคือ FA หรือไฟล์แอตทริบิวต์ ซึ่งเป็นโปรแกรมที่ใช้สำหรับปรับเปลี่ยนแอตทริบิวต์ของไฟล์ เพราะการระวังรักษาความปลอดภัยระดับแรกก็คือ การปรับเปลี่ยนแอตทริบิวต์ของไฟล์ซึ่งทำหน้าที่ควบคุมคุณลักษณะของไฟล์ โดยในระบบดำเนินการดอสมีแอตทริบิวต์ควบคุมอยู่หลายส่วน เช่น การกำหนดให้อ่านไฟล์ได้อย่างเดียว (Read-Only) การซ่อนไฟล์ (Hidden) เป็นต้น

อีกสองโปรแกรมที่เหลือเป็นโปรแกรมเกี่ยวกับการเข้ารหัสข้อมูล (Encyption) ในไฟล์ การเข้ารหัสข้อมูลจะทำให้ข้อมูลเปลี่ยนแปลงไปและทำให้ไม่สามารถอ่านหรือดูข้อมูลได้ โปรแกรม ENC เป็นโปรแกรมเข้ารหัสธรรมดา ซึ่งไฟล์ข้อมูลที่ถูกเข้ารหัสแล้วก็สามารถแปลงคืนกลับมาได้ดังเดิม ส่วนโปรแกรม FSEC เป็นโปรแกรมเข้ารหัสข้อมูลที่ต้องใช้รหัสผ่าน (Password) ซึ่งในการแปลงข้อมูลคืนกลับจะต้องใช้รหัสผ่านด้วย จึงสามารถแปลงกลับได้

โปรแกรม FA (File Attribute)

การทำงานของโปรแกรม

โปรแกรม FA ใช้สำหรับเปลี่ยนแปลงแอตทริบิวต์ ซึ่งเป็นการรักษาความปลอดภัยไฟล์ข้อมูลระดับแรกด้วยการทำงานของโปรแกรม FA สามารถปรับเปลี่ยนแอตทริบิวต์ของไฟล์ เช่น การซ่อนไฟล์ไม่ให้ผู้อื่นสามารถมองเห็นได้ หรือสับเปลี่ยนแอตทริบิวต์ของไฟล์ให้อ่านได้เพียงอย่างเดียว เพื่อป้องกันการแก้ไขหรือป้องกันการถูกลบทิ้ง

ความจริงแล้วในดอสเองก็มีโปรแกรมสำหรับทำหน้าที่นี้อยู่แล้ว คือ โปรแกรม ATTRIB โดยเฉพาะในเอ็มเอสดอสรุ่น 6.0 นั้น โปรแกรม ATTRIB มีขีดความสามารถสูงมาก โดยนอกจากจะสามารถใช้ปรับเปลี่ยนแอตทริบิวต์ของไฟล์ได้ทุกแอตทริบิวต์แล้ว ยังสามารถระบุให้ปรับเปลี่ยนแอตทริบิวต์ของไฟล์ในไดเรคทอรีย่อยที่อยู่ระดับลึกลงไปได้อีกด้วย อย่างไร

ก็ตามยังมีผู้ใช้อีกหลายคนที่ไม่ได้เปลี่ยนรุ่นของดอสตามรุ่นใหม่ ดอสในรุ่น 3.0 จนถึง 3.3 ซึ่งถึงแม้จะมีโปรแกรม ATTRIB ติดมาให้ก็ตาม แต่โปรแกรม ATTRIB ที่มีให้นั้นก็มีข้อจำกัดอยู่โดยใช้เปลี่ยนแอตทริบิวต์ได้เฉพาะแอตทริบิวต์อ่านอย่างเดียวเท่านั้น ซึ่งหากเป็นกรณีนี้โปรแกรม FA สามารถจะช่วยงานได้

รูปแบบการใช้งานโปรแกรม

โปรแกรม FA มีลักษณะการใช้งานง่าย ๆ ดังนี้

```
fa filename.ext [/r/h/s]
```

การใช้งานทำโดยเรียกชื่อโปรแกรม FA ตามด้วยชื่อไฟล์พร้อมระบุนามสกุล แล้วตามด้วยแอตทริบิวต์ที่ต้องการ ชื่อไฟล์และนามสกุลสามารถระบุชื่อเดียวหรือเป็นกลุ่มได้ด้วยอักษร wildcard ส่วนแอตทริบิวต์สามารถเปลี่ยนได้ 3 แอตทริบิวต์ผสมกันได้คือ อ่านอย่างเดียว (/r) ซ่อนไฟล์ (/h) และไฟล์ของระบบ (/s) ซึ่งหากไม่มีการระบุแอตทริบิวต์ โปรแกรมจะแอตทริบิวต์ของไฟล์ให้เป็นแบบปกติ

ตัวอย่าง

```
C:\ >fa c:\micro/*.c /r
```

จากตัวอย่างนี้เป็นการเปลี่ยนแอตทริบิวต์ของไฟล์ทุกไฟล์ในไดเรคทอรี micro ที่มีนามสกุลเป็น .c ให้มีแอตทริบิวต์เป็นอ่านอย่างเดียว

```
C:>\fa member.dbf /r/h/s
```

ตัวอย่างนี้เป็นการเปลี่ยนแอตทริบิวต์ของไฟล์ member.dbf ให้เป็นไฟล์ที่อ่านได้อย่างเดียว ซ่อนไฟล์ และให้กลายเป็นไฟล์ของระบบด้วย ซึ่งมีผล

ลิสติงที่ 1 โปรแกรม FA (File Attribute)

```
1  /*
2
3     Title          :   FA.EXE   (File Attribute)
4     Purpose        :            Changing of file attribute byte with thsi utility
5                                 you can make files hidden files, system files or
6                                 read-only  or any combination of these.
7
8     Requirements   :            The program will work on MS-DOS and PC-DOS computer
9                                 using version 2.0 or  later
10
11 */
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <io.h>
16 #include <string.h>
17 #include <dir.h>
18 #include <dos.h>
19 #include <conio.h>
20 #include <ctype.h>
21
22 /*
23 The two internal functions used in this program. Neither one of
24 the functions accepts arguments nor returns values.
25 */
26
27 void attb (void);
28 void attbaction (void);
29
30 /*
31 Global variables used throughout the program.
32 */
33
34 char *workstr;
35 char searchstr [MAXPATH], attrib [7], option [128],
36    name [MAXPATH], pathname [MAXPATH], drive [MAXDRIVE],
37    subdir [MAXDIR], file [MAXFILE], ext [MAXEXT];
38
39 int i, done, answer,
40    ok, attribnum, numofargs;
41
42 struct ffblk dta;
43
44 /*
45 FA works like many of utilitys. It searchs through whatever subdirectory
46 you specify and chang the file attribute byte of that file.
47
48 The first argument is copied to "searchstr" and this is later broken
49 into usable components by the utility.
50
51 CALLED FROM: (DOS)
52
53 CALLS:     attb
54
55 */
56
57 main (int argc, char *argv[])
58 {
59   printf ("\nFA-File Attribute, TC Edition.\n");
60
61   numofargs = argc - 1;
62   if (numofargs < 1)
63     {
64     printf ("\nUseage: C:\>fa filename.ext [/r/h/s]\n");
65     printf ("\nfor normal attribute: C:\>fa filename.ext\n");
```

ทำให้ไม่สามารถมองเห็นไฟล์นี้ และไม่สามารถแก้ไขและเปลี่ยนแก้ไขได้

```
C: \ >fa member.dbf
```

ตัวอย่างนี้เป็นการเปลี่ยนแอตทริบิวต์ของไฟล์ member.dbf ให้เป็นปกติ

```
C: \ >fa
```

หากเรียกโปรแกรมเฉย ๆ และโปรแกรมจะแสดงข้อความช่วยแสดงถึงการใช้งานโปรแกรม

เทคนิคโปรแกรม

โปรแกรม FA จะใช้ฟังก์ชันของเทอร์โบซี คือฟังก์ชัน __chmod ในการปรับเปลี่ยนแอตทริบิวต์ของไฟล์ที่ระบุ โดยการใช้ทำงานฟังก์ชันนี้ต้องบอกพารามิเตอร์ 3 ตัวด้วยกัน ซึ่งพารามิเตอร์ตัวแรกก็คือชื่อไฟล์ที่ต้องการปรับเปลี่ยนแอตทริบิวต์ พารามิเตอร์ตัวถัดไปเป็นจำนวนเต็ม ซึ่งหากมีค่าเป็น 1 จะเป็นการบอกให้ฟังก์ชัน __chmod เปลี่ยนแอตทริบิวต์ของไฟล์ตาม พารามิเตอร์ตัวสุดท้ายที่ค่าของไบต์แอตทริบิวต์ แต่หากมีค่าเป็น 0 จะเป็นเพียงการอ่านเอาค่าแอตทริบิวต์ของไฟล์นั้นออกมา ฉย ๆ ฟังก์ชันนี้ถูกประกาศอยู่ในไฟล์ O.H ส่วนค่าคงที่ของแอตทริบิวต์คือ

```
FA__RDONLY 1
FA__HIDDEN 2
FA__SYSTEM 4
```

ประกาศอยู่ในไฟล์ DOS.H

การทำงานของโปรแกรม FA จะมาจากการตรวจสอบชื่อไฟล์ที่ระบุและแอตทริบิวต์ที่ต้องการปรับเปลี่ยน จาก

ลิสติงที่ 1 (ต่อ)

```
66      return (1);
67      }
68    else
69      {
70      strcpy (searchstr, argv[1]);
71      fnsplit (searchstr, drive, subdir, file, ext);
72      sprintf (pathname, "%s%s", drive, subdir);
73      attribnum = 32;
74      }
75
76    for (i = 2; i <= numofargs; i++)
77      strcat (option, argv [i]);
78
79    workstr = strpbrk (option,"Rr");
80    if (workstr != NULL)
81      {
82      strcat (attrib, "R");
83      attribnum += FA_RDONLY;
84      }
85
86    workstr = strpbrk (option,"Hh");
87    if (workstr != NULL)
88      {
89      strcat (attrib, "H");
90      attribnum += FA_HIDDEN;
91      }
92
93    workstr = strpbrk (option,"Ss");
94    if (workstr != NULL)
95      {
96      strcat (attrib, "S");
97      attribnum += FA_SYSTEM;
98      }
99
100   attb ();
101   return (0);
102 }
103
104 /* This function searchs for files that math the "searchstr".
105 Once a math is found, several commands in the procedure create
106 a destination name which include the "pathname" specified at
107 the command line. "attbaction" is then called to ask user if
108 really wants to carry out the action.
109
110 CALLED FROM: main
111
112 CALLED:     attbaction
113 */
114
115 void attb (void)
116 {
117   done = findfirst (searchstr, &dta, 47);
118
119   while (!done)
120     {
121     strcpy (name, pathname);
122     strcat (name,dta.ff_name);
123     strupr (name);
124
125     attbaction ();
126     done = findnext (&dta);
127     }
128   printf ("\nNo more matching files.\n");
129 }
130
131 /*
132 "attbaction" displays a message on the screen that prompts the user
```

นั้นจึงเตรียมชื่อไฟล์ให้อยู่ในรูปที่จะใช้ในการค้นหา แล้วทำการค้นหาไฟล์ด้วยฟังก์ชัน findfirst ( ) และ findnext ( ) (เป็นฟังก์ชันที่เราเคยใช้งานมาแล้วในโปรแกรมยูทิลิตี้ที่ผ่านมา) หากพบไฟล์ตามที่ระบุก็จะทำการปรับเปลี่ยนแอตทริบิวต์ด้วยฟังก์ชัน __chmod ตามที่กล่าวมาแล้ว

โปรแกรม ENC (Enerypt)

การทำงานของโปรแกรม

วิธีการรักษาความปลอดภัยไฟล์ข้อมูลอีกวิธีหนึ่งก็คือ การเข้ารหัสข้อมูล การเข้ารหัสข้อมูลคือเปลี่ยนแปลงข้อมูลให้อยู่ในรูปใหม่ที่ไม่สามารถจะอ่านหรือพิจารณาดูข้อมูลเดิมได้ ซึ่งในปัจจุบันก็มีโปรแกรมยูทิลิตี้หลายโปรแกรมสำหรับช่วยงานในด้านนี้ โปรแกรม ENC ก็เป็นโปรแกรมหนึ่งซึ่งสามารถจะทำการเข้ารหัสข้อมูลได้อย่างรวดเร็วด้วยวิธีการง่าย ๆ

การเข้ารหัสข้อมูลของโปรแกรม ENC นี้จะใช้ตำแหน่งของข้อมูลในบัฟเฟอร์มาใช้ช่วยในการเข้ารหัส ซึ่งความจริงแล้ววิธีการเข้ารหัสข้อมูลมีหลายวิธี แต่วิธีการเข้ารหัสข้อมูลในโปรแกรม ENC เป็นวิธีการที่เรียบง่ายวิธีหนึ่ง โดยหลังจากการเข้ารหัสข้อมูลไฟล์ไปแล้ว ถ้าต้องการจะแปลงข้อมูลกลับคืน ก็สามารถทำได้โดยการเรียกผ่านโปรแกรม ENC อีกครั้งหนึ่ง จะได้ข้อมูลกลับคืนดังเดิม

ดังนั้นข้อควรระวังในการใช้งานโปรแกรมนี้ก็คือ หลังจากการเข้ารหัสไฟล์แล้วไม่ควรจะทิ้งโปรแกรมนี้เอาไว้ในเครื่อง เพราะจะทำให้ผู้อื่นสามารถแปลงข้อมูลกลับได้

ลิสติงที่ 1 (ต่อ)

```
133 of the program for Y(es) or N9o) answer. Depending on the answer
134 the procedure will either change the file's attribute byte or ignore it.
135 */
136
137 void attbaction (void)
138 {
139   printf (" Change the attribute of %-12s to",name);
140   if (strlen(attrib) < 1)
141     {
142     strcpy(attrib,"NORMAL");
143     printf (" %s ",attrib);
144     }
145   else
146     printf (" %-3s? ", attrib);
147
148   ok = _chmod (name, 1, attribnum);
149   if (ok)
150     printf ("- done.\n");
151   else
152     printf ("-not altered. ERROR !\n");
153 }
154
155
```

ลิสติงที่ 2 โปรแกรม ENC (Enerypt)

```
1 /*
2
3     Title          :   ENC.EXE   (Encrypt data file)
4     Purpose                      Encrypts of data files on a disk. All bytes in the
5                    :            file are encrypted with an algorithm and can be
6                                 brought back by running them through this program a
7                                 a second time.
8
9     Requirements   :             The program will work on MS-DOS and PC-DOS computer
10                                using version 2.0 or later
11
12 */
13
14 #include <stdio.h>
15 #include <string.h>
16 #include <alloc.h>
17 #include <dir.h>
18 #include <io.h>
19
20 /*
21 Function other than "main" used in this utility
22 */
23
24 void encrypt (int loopnum);
25
26 /*
27 Global variables used throughout the program.
28 */
29
30 FILE *secret;
31 int  handle, numtoget, loopnum;
32 long size, filepos, numread;
33 char filestr [MAXPATH];
```

ข้อดีของโปรแกรมนี้คือ การสร้างไฟล์ข้อมูลที่ถูกเข้ารหัสจะใช้วิธีการเขียนทับลงไปในไฟล์เดิม ซึ่งจะเป็นวิธีการที่ปลอดภัยกว่าการสร้างไฟล์ใหม่ที่ได้จากการเข้ารหัสแล้วลบไฟล์เดิมทิ้ง เพราะการทำเช่นนี้ยังสามารถเรียกไฟล์ที่ลบทิ้งกลับคืนมาได้ นอกจากนี้การเข้ารหัสของโปรแกรมก็ไม่ได้ใช้รหัสผ่าน ดังนั้นจึงสามารถแปลงข้อมูลกลับได้เสมอ โดยไม่ต้องกังวลเรื่องของการที่จะต้องจดจำรหัสผ่าน

รูปแบบของการใช้งาน

```
enc filename.ext
```

เรียกชื่อโปรแกรมแล้วตามด้วยชื่อไฟล์ที่ต้องการเข้ารหัส ซึ่งในการระบุชื่อไฟล์สามารถบอกชื่อพาธหรือไดเรคทอรีที่ไฟล์นั้นเก็บอยู่ได้ด้วย จากนั้นโปรแกรม ENC จะอ่านข้อมูลในไฟล์มาเข้ารหัสแล้วเรียกกลับลงไปในไฟล์เดิม

ตัวอย่างการใช้งาน

```
C: \ >enc C: \ micro \ fa.c
```

ตัวอย่างนี้เป็นการเข้ารหัสข้อมูลของไฟล์โปรแกรมต้นฉบับภาษาซีชื่อ fa เพื่อป้องกันผู้อื่นมาอ่าน จะเห็นว่าการใช้งานโปรแกรม enc การระบุชื่อไฟล์สามารถบอกถึงชื่อไดเรคทอรีที่ใช้เก็บไฟล์ได้ด้วย จากตัวอย่างนี้หากต้องการไฟล์โปรแกรมต้นฉบับกลับคืนก็สามารถทำได้ โดยการเรียกโปรแกรม enc แล้วตามชื่อไฟล์นี้อีกครั้งหนึ่ง

```
B: \ >enc
```

หากเรียกชื่อโปรแกรมอย่างเดียวจะแสดงวิธีการใช้โปรแกรมให้

ลิสติงที่ 2 (ต่อ)

```
34  unsigned char *buffer;
35
36  /*
37  The "main" loop sets up the buffer which used later in the
38  program and getsb command line arguments from the DOS command line.
39  Any errors are reported to user and execution of the program ends.
40
41  CALLED FROM:  (DOS)
42
43  CALLS:        encript
44  */
45
46  main (int argc, char *argv[])
47  {
48    printf ("\nENC-Encript Files, TC-Edition\n");
49
50    buffer = (unsigned char *) calloc (32767,1);
51    if (buffer  == NULL)
52      {
53      printf ("\nNot enough memory to run application. Please remove");
54      printf ("\nall memory resident program and try again\n");
55      return (1);
56      }
57
58    if (argc >1)
59      strcpy (filestr, argv[1]);
60    else
61      {
62      printf ("\nUsaged: C:\>enc filename.ext\n");
63      return (2);
64      }
65
66    secret = fopen (filestr,"r+b");
67    if (secret == NULL)
68      {
69      printf ("\nThe file was not found. Please check the spelling.\n");
70      return (3);
71      }
72
73    handle = fileno (secret);
74    size = filelength (handle);
75
76    if (size <= 32767)
77      numtoget = size;
78    else
79      numtoget = 32767;
80    numread = 0;
81
82    do
83      {
84      filepos = ftell (secret);
85      fseek (secret, filepos,0);
86      _read (handle, buffer, numtoget);
87      numread += numtoget;
88
89      encrypt (numtoget);
90
91      fseek (secret, filepos, 0);
92      _write (handle, buffer, numtoget);
93
94      if ((numread + numtoget) > size)
95        numtoget = size - numread;
96      }
97    while (numtoget != 0);
98
99    fclose (secret);
100   return (0);
```

เทคนิคโปรแกรม

การทำงานของโปรแกรมนี้สามารถทำงานได้อย่างรวดเร็ว เพราะใช้วิธีการอ่าน/เขียนข้อมูลจากไฟล์ทีละ 32 กิโลไบต์ โดยการใช้ฟังก์ชัน __read ( ) และ __write ( ) ซึ่งสามารถจะเปลี่ยนเป็นฟังก์ชัน read ( ) และ write ( ) ได้ หากต้องการจะย้ายโปรแกรมไปใช้งานบนยูนิกซ์ (UNIX)

```
int __read (int handle, void
*buffer, int numofbytes);
```

การใช้งานฟังก์ชัน __read ( ) และ write ( ) นั้นต้องการพารามิเตอร์ 3 ตัว โดยพารามิเตอร์ตัวแรกคือค่าตัวเลขลำดับของการเปิดไฟล์ (File Handle) ซึ่งความจริงแล้วหากเปิดไฟล์ด้วยฟังก์ชัน open ( ) หรือ __open ( ) ก็จะได้ค่าตัวเลขทันที แต่ในโปรแกรมใช้วิธีการเปิดไฟล์ด้วยฟังก์ชัน open ( ) ซึ่งส่งพอยน์เตอร์ของไฟล์มาให้ แต่ก็สามารถจะเปลี่ยนให้เป็นค่าลำดับการเปิดไฟล์ได้ด้วยฟังก์ชัน fileno ( )

การทำงานโปรแกรมจะเริ่มต้นจากการตรวจสอบไฟล์ข้อมูลที่ต้องการเข้ารหัส ซึ่งหากไฟล์ดังกล่าวไม่มีจริงก็จะจบการทำงาน แต่หากเปิดไฟล์ได้ก็จะอ่านข้อมูลจากไฟล์มาทีละ 32 กิโลไบต์ แล้วทำการเข้ารหัส จากนั้นเขียนข้อมูลกลับยังตำแหน่งเดิม

วิธีการการเข้ารหัสของโปรแกรมนี้จะใช้ตำแหน่งของข้อมูลในบัฟเฟอร์เป็นส่วนสำคัญในการเข้ารหัส โดยการนำข้อมูลมาเอ็กซ์คลูซีฟออร์ (Exclusive Or) ส่วนวิธีการเข้ารหัสจะทิ้งไว้ให้ผู้อ่านพิจารณาเอง

ลิสติงที่ 2 (ต่อ)

```
} /* end of main */

/*
This function encrypts all bytes which the main has placed in
"buffer". Notice that a formula is used and that space are skiped.
This is  done so that there are no repeating characters in the file.
The position within the buffer is also used as part of the encryption
to decrease the likelihood of repeating character.

CALLED FROM:  main

CALLS:       (none)
*/

void encrypt (int loopnum)
{
  int i;

  i = 0;
  do
    {
    buffer[i] = buffer[i] ^ ((255 - (i & 254)) | (127 ^ i));
    i++;
    }
  while (i != loopnum);
}

```

ลิสติงที่ 3 โปรแกรม FSEC-File Secure

```
/*

    Title          :   FSEC.EXE   (File Secret)
    Purpose        :          Encrypts of data files on a disk. All bytes in the
                              file are encrypted with an algorithm and can be
                              brought back by running them through this program a
                              a second time. Unlike ENC, this program requires
                              that a password be entered. The password adds an
                              extra level of protection that ENC does not.

    Requirements   :          The program will work on MS-DOS and PC-DOS computer
                              using version 2.0 or later

*/

#include <stdio.h>
#include <string.h>
#include <alloc.h>
#include <dir.h>
#include <io.h>

/*
Function other than "main" used in this utility
*/

void encrypt (int loopnum);

/*
Global variables used throughout the program.
```

โปรแกรม FSEC - File Secure

การทำงานของโปรแกรม

โปรแกรม FSEC จะมีการทำงานเช่นเดียวกับโปรแกรม ENC คือใช้สำหรับเข้ารหัสไฟล์ข้อมูล แต่ต้องระบุรหัสผ่านเพื่อใช้ในการเข้ารหัสข้อมูล ซึ่งจะเป็นการป้องกันมิให้มีการแปลงข้อมูลกลับโดยผู้อื่น เพราะหากมีการใช้โปรแกรม FSEC เพื่อจะแปลงข้อมูลกลับ จะต้องระบุรหัสผ่านให้ถูกต้องจึงจะสามารถแปลงข้อมูลกลับได้ถูกต้องดังเดิม

รูปแบบการใช้งาน

```
fsec filename.ext password
```

การใช้งานโปรแกรม FSEC จะคล้ายกับโปรแกรม ENC คือตามหลังชื่อโปรแกรมจะเป็นชื่อไฟล์ที่ต้องการเข้ารหัส (password) ซึ่งรหัสผ่านสามารถจะใส่ข้อความได้ยาวถึง 37 ตัวอักษร (ไม่นับช่องว่าง)

ส่วนการแปลงข้อมูลกลับทำเช่นเดียวกับการเข้ารหัส แต่จะต้องใส่รหัสผ่านให้ถูกต้อง ข้อควรระวังอันหนึ่งก็คือ หากได้ทำการเข้ารหัสไฟล์เอาไว้แล้วควรจะเก็บสำรองไฟล์ที่เข้ารหัสเอาไว้ด้วย และจะต้องจำรหัสผ่านให้ได้ที่ต้องทำเช่นนั้นเพราะว่า หากมีผู้ที่จะพยายามแปลงข้อมูลกลับโดยการทดลองใส่รหัสผ่านดู ไฟล์ข้อมูลจะถูกเข้ารหัสอีกครั้ง ทำให้ข้อมูลที่มีอยู่ในไฟล์ถูกเปลี่ยนแปลงไปอีกครั้ง (หรือจำรหัสผ่านผิดก็จะได้ผลเช่นเดียวกัน) ดังนั้นเพื่อความแน่ใจจึงควรจะมีสำรองไฟล์ที่เข้ารหัสครั้งแรกเอาไว้ด้วย

ลิสติงที่ 3 (ต่อ)

```
30 */
31
32 FILE *secret;
33 int  handle, numtoget, loopnum;
34 long  size, filepos, numread;
35 char  filestr [MAXPATH], password  [128];
36 unsigned char *buffer ;
37
38 /*
39 The "main" loop sets up the buffer which used later in the
40 program and getsb command line arguments from the DOS command line.
41 Any errors are reported to user and execution of the program ends.
42
43 CALLED FROM:  (DOS)
44
45 CALLS:·      encript
46 */
47
48 main (int argc, char *argv[])
49 {
50   int i;
51
52   printf ("\nFSEC-File Secret, TC-Edition\n");
53
54   buffer = (unsigned char *) calloc (32767,1);
55   if (buffer  == NULL)
56     {
57     printf ("\nNot enough memory to run application. Please remove");
58     printf ("\nall memory resident program and try again\n");
59     return (1);
60     }
61
62   if (argc > 2)
63     {
64     strcpy (filestr, argv[1]);
65     for (i = 2; i <= (argc -1); i++)
66       strcat (password, argv[i]);
67     }
68   else
69     {
70     printf ("\nUsaged: C:\>enc filename.ext\n");
71     return (2);
72     }
73
74   secret = fopen (filestr,"r+b");
75   if (secret == NULL)
76     {
77     printf ("\nThe file was not found. Please check the spelling.\n");
78     return (3);
79     }
80
81   handle = fileno (secret);
82   size = filelength (handle);
83
84   if (size <= 32767)
85     numtoget = size;
86   else
87     numtoget = 32767;
88   numread = 0;
89
90   do
91     {
92     filepos = ftell (secret);
93     fseek (secret, filepos,0);
94     _read (handle, buffer, numtoget);
95     numread += numtoget;
96
```

ลิสติงที่ 3 (ต่อ)

```
97      encrypt (numtoget);
98
99      fseek (secret, filepos, 0);
100     _write (handle, buffer, numtoget);
101
102     if ((numread + numtoget) > size)
103       numtoget = size - numread;
104     }
105   while (numtoget != 0);
106
107   fclose (secret);
108   return (0);
109 } /* end of main */
110
111 /*
112 This function encrypts all bytes which the main has placed in
113 "buffer". Notice that a formula is used and that space are skiped.
114 This is  done so that there are no repeating characters in the file.
115 The position within the buffer is also used as part of the encryption
116 to decrease the likelihood of repeating character.
117
118 CALLED FROM:  main
119
120 CALLS:       (none)
121 */
122
123 void encrypt (int loopnum)
124 {
125   int passindx, j;
126
127   j = 0;
128   passindx = 0;
129   do
130     {
131     buffer[j] = buffer[j] ^ ((255 - (j & 254)) | (127 ^ j));
132     passindx++;
133     if (passindx > 37)
134       passindx =0;
135     buffer [j] = buffer[j] ^ password [passindx];
136     j++;
137     }
138   while (j != loopnum);
139 }
140
```

ตัวอย่าง

```
B: \ > fsec love.doc romance forever
```

เป็นการเข้ารหัสไฟล์จดหมายรัก
ชื่อ love ด้วยรหัสผ่านคือข้อความ
romance forever

ให้สังเกตว่าข้อความที่ใช้เป็นรหัส
ผ่านสามารถจะมีช่องว่างระหว่างคำได้
ซึ่งช่องว่างนี้จะถูกตัดออก

```
C: \ > fsec c: \ micro \ enc.c security
```

ตัวอย่างเป็นการเข้ารหัสผ่านไฟล์
โปรแกรมต้นฉบับภาษาซีชื่อ enc โดย
ใช้รหัสผ่าน security

```
C: \ > fsec test.txt
```

หากเรียกโปรแกรม fsec ตาม
ด้วยชื่อไฟล์แต่ไม่บอกรหัสผ่าน โปรแกรม
จะไม่ทำงาน และจะแสดงวิธีการใช้งาน
โปรแกรมมาให้

```
C: \ > fsec
```

หากเรียกชื่อโปรแกรมอย่างเดียว
โปรแกรมจะแสดงวิธีการใช้งานให้

เทคนิคโปรแกรม

เทคนิคการทำงานโปรแกรม FSE
ก็เช่นเดียวกับ ENC แต่ในการเข้ารหัส
ข้อมูลจะนำเอารหัสผ่านมาร่วมผสมกา
เข้ารหัสด้วย